

Micah Lee, Glenn Greenwald, Morgan Marquis-Boire

July 2 2015, 10:42 a.m.

Second in a series. Part 1 [here](#).

The sheer quantity of communications that XKEYSCORE processes, filters and queries is stunning. Around the world, when a person gets online to do anything — write an email, post to a social network, browse the web or play a video game — there’s a decent chance that the Internet traffic her device sends and receives is getting collected and processed by one of XKEYSCORE’s hundreds of servers scattered across the globe.

In order to make sense of such a massive and steady flow of information, analysts working for the National Security Agency, as well as partner spy agencies, have written thousands of snippets of code to detect different types of traffic and extract useful information from each type, according to documents dating up to 2013. For example, the system automatically detects if a given piece of traffic is an email. If it is, the system tags if it's from Yahoo or Gmail, if it contains an airline itinerary, if it's encrypted with PGP, or if the sender's language is set to Arabic, along with myriad other details.

This global Internet surveillance network is powered by a somewhat clunky piece of software running on clusters of Linux servers. Analysts access XKEYSCORE's web interface to search its wealth of private information, similar to how ordinary people can search Google for public information.

Based on documents provided by NSA whistleblower Edward Snowden, *The Intercept* is shedding light on the inner workings of XKEYSCORE, one of the most extensive programs of mass surveillance in human history.

How XKEYSCORE works under the hood

It is tempting to assume that expensive, proprietary operating systems and software must power XKEYSCORE, but it actually relies on an entirely open source stack. In fact, according to an analysis of an XKEYSCORE manual for new systems administrators from the end of 2012, the system may have design deficiencies that could leave it vulnerable to attack by an intelligence agency insider.

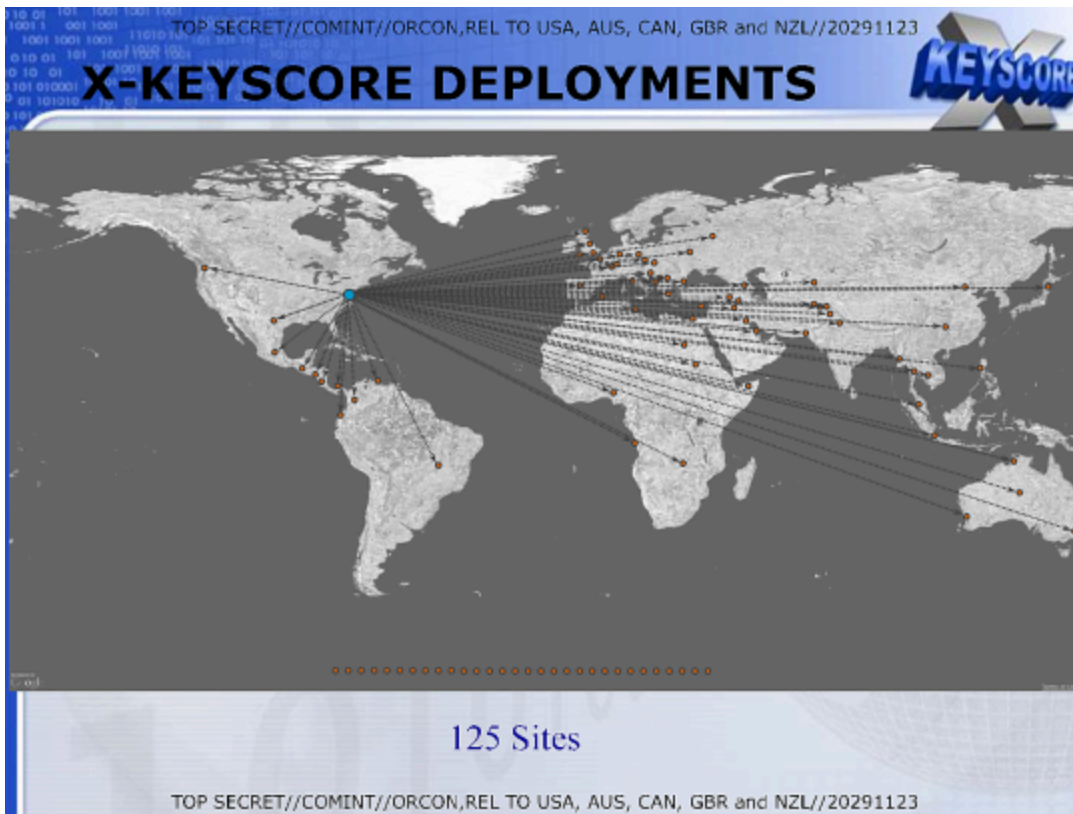
XKEYSCORE is a piece of Linux software that is typically deployed on Red Hat servers. It uses the Apache web server and stores collected data in MySQL databases. File systems in a cluster are handled by the

NFS distributed file system and the autofs service, and scheduled tasks are handled by the cron scheduling service. Systems administrators who maintain XKEYSCORE servers use SSH to connect to them, and they use tools such as rsync and vim, as well as a comprehensive command-line tool, to manage the software.

John Adams, former security lead and senior operations engineer for Twitter, says that one of the most interesting things about XKEYSCORE's architecture is "that they were able to achieve so much success with such a poorly designed system. Data ingest, day-to-day operations, and searching is all poorly designed. There are many open source offerings that would function far better than this design with very little work. Their operations team must be extremely unhappy."

Analysts connect to XKEYSCORE over HTTPS using standard web browsers such as Firefox. Internet Explorer is not supported. Analysts can log into the system with either a user ID and password or by using public key authentication.

As of 2009, XKEYSCORE servers were located at more than 100 field sites all over the world. Each field site consists of a cluster of servers; the exact number differs depending on how much information is being collected at that site. Sites with relatively low traffic can get by with fewer servers, but sites that spy on larger amounts of traffic require more servers to filter and parse it all. XKEYSCORE has been engineered to scale in both processing power and storage by adding more servers to a cluster. According to a 2009 document, some field sites receive over 20 terabytes of data per day. This is the equivalent of 5.7 million songs, or over 13 thousand full-length films.



This map from a 2009 top-secret presentation does not show all of XKEYSCORE's field sites.

When data is collected at an XKEYSCORE field site, it is processed locally and ultimately stored in MySQL databases at that site.

XKEYSCORE supports a federated query system, which means that an analyst can conduct a single query from the central XKEYSCORE website, and it will communicate over the Internet to all of the field sites, running the query everywhere at once.

There might be security issues with the XKEYSCORE system itself as well. As hard as software developers may try, it's nearly impossible to write bug-free source code. To compensate for this, developers often rely on multiple layers of security; if attackers can get through one layer, they may still be thwarted by other layers. XKEYSCORE appears to do a bad job of this.

When systems administrators log into XKEYSCORE servers to configure them, they appear to use a shared account, under the name "oper." Adams notes, "That means that changes made by an administrator cannot be logged." If one administrator does something

malicious on an XKEYSCORE server using the “oper” user, it’s possible that the digital trail of what was done wouldn’t lead back to the administrator, since multiple operators use the account.

There appears to be another way an ill-intentioned systems administrator may be able to cover their tracks. Analysts wishing to query XKEYSCORE sign in via a web browser, and their searches are logged. This creates an audit trail, on which the system relies to assure that users aren’t doing overly broad searches that would pull up U.S. citizens’ web traffic. Systems administrators, however, are able to run MySQL queries. The documents indicate that administrators have the ability to directly query the MySQL databases, where the collected data is stored, apparently bypassing the audit trail.

AppIDs, fingerprints and microplugins

Collecting massive amounts of raw data is not very useful unless it is collated and organized in a way that can be searched. To deal with this problem, XKEYSCORE extracts and tags metadata and content from the raw data so that analysts can easily search it.

This is done by using dictionaries of rules called appIDs, fingerprints and microplugins that are written in a custom programming language called GENESIS. Each of these can be identified by a unique name that resembles a directory tree, such as “mail/webmail/gmail,” “chat/yahoo,” or “botnet/blackenergybot/command/flood.”

One document detailing XKEYSCORE appIDs and fingerprints lists several revealing examples. Windows Update requests appear to fall under the “update_service/windows” appID, and normal web requests fall under the “http/get” appID. XKEYSCORE can automatically detect Airblue travel itineraries with the “travel/airblue” fingerprint, and

iPhone web browser traffic with the “browser/cellphone/iphone” fingerprint.

PGP-encrypted messages are detected with the “encryption/pgp/message” fingerprint, and messages encrypted with Mojahdeen Secrets 2 (a type of encryption popular among supporters of al Qaeda) are detected with the “encryption/mojaheden2” fingerprint.

When new traffic flows into an XKEYSCORE cluster, the system tests the intercepted data against each of these rules and stores whether the traffic matches the pattern. A slideshow presentation from 2010 says that XKEYSCORE contains almost 10,000 appIDs and fingerprints.

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

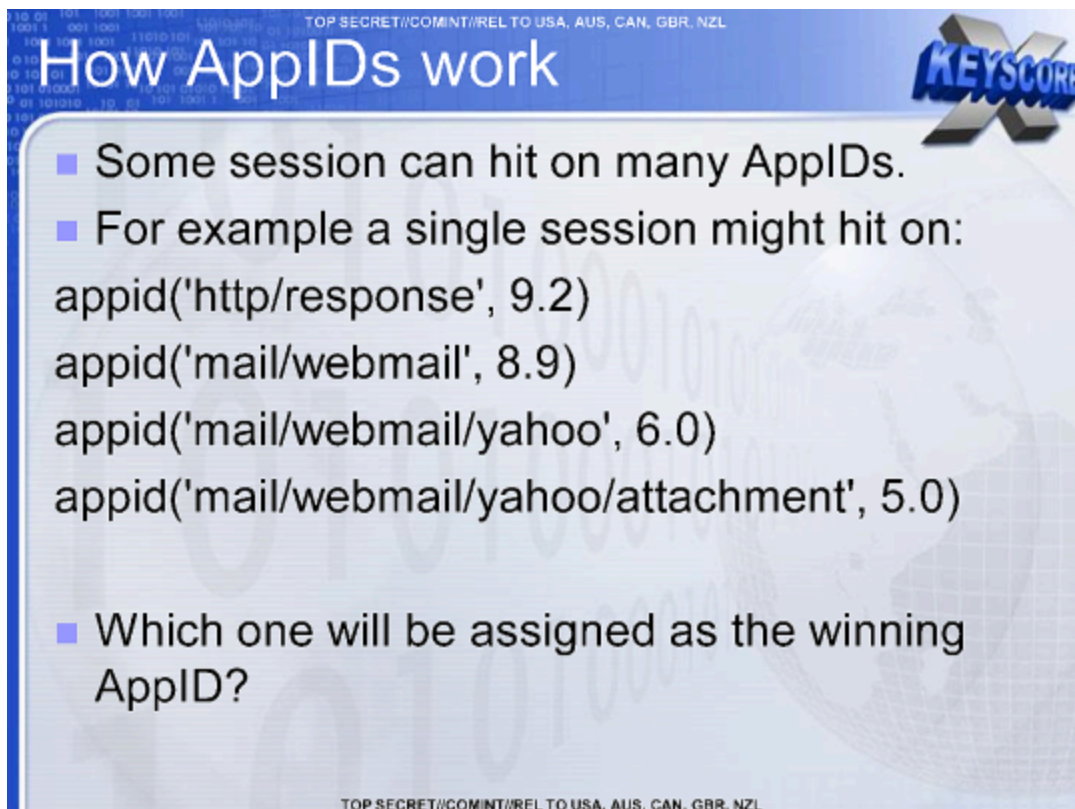
“There’s an App for that!”

- There are currently almost 10,000 AppIDs and Fingerprints in X-KEYSCORE – the full list is available from the NSA XKS Home Page
- Odds are there may already be a fingerprint for the traffic you’re interested in.
- If not you can easily create your own!

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

AppIDs are used to identify the protocol of traffic being intercepted, while fingerprints detect a specific type of content. Each intercepted stream of traffic gets assigned up to one appID and any number of fingerprints. You can think of appIDs as categories and fingerprints as tags.

If multiple appIDs match a single stream of traffic, the appID with the lowest “level” is selected (appIDs with lower levels are more specific than appIDs with higher levels). For example, when XKEYSCORE is assessing a file attachment from Yahoo mail, all of the appIDs in the following slide will apply, however only “mail/webmail/yahoo/attachment” will be associated with this stream of traffic.



The slide is titled "How AppIDs work" in a large, bold, white font. In the top right corner, there is a logo for "XKEYSCORE" with a large "X" and the word "KEYSCORE" in blue. The slide contains a list of four appIDs with their levels in parentheses, each preceded by a blue square bullet point. The appIDs are: "http/response" (9.2), "mail/webmail" (8.9), "mail/webmail/yahoo" (6.0), and "mail/webmail/yahoo/attachment" (5.0). Below these, another blue square bullet point asks, "Which one will be assigned as the winning AppID?". The slide has a blue header and footer with the text "TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL".

How AppIDs work

- Some session can hit on many AppIDs.
- For example a single session might hit on:
appid('http/response', 9.2)
appid('mail/webmail', 8.9)
appid('mail/webmail/yahoo', 6.0)
appid('mail/webmail/yahoo/attachment', 5.0)
- Which one will be assigned as the winning AppID?

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

To tie it all together, when an Arabic speaker logs into a Yahoo email address, XKEYSCORE will store “mail/yahoo/login” as the associated appID. This stream of traffic will match the “mail/arabic” fingerprint (denoting language settings), as well as the “mail/yahoo/ymbm” fingerprint (which detects Yahoo browser cookies).

TOP SECRET//COMINT//REL TO USA, FVEY

Example

KEYSCORE

```
appid('mail/yahoo', 9.0) = 'Host: mail.yahoo';
appid('mail/yahoo/login', 8.0) = 'Host: mail.yahoo' and '/login';

fingerprint('mail/arabic') = 'mail' and /language[:=] ?ar/;
fingerprint('mail/yahoo/ymbm') = 'Host: mail.yahoo' and 'YMBM='c;
```

```
GET /login.html HTTP/1.1
Referer: http://us.f359.mail.yahoo.com/ym/ShowLetter
Accept-Language: ar
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: mail.yahoo.com
Connection: Keep-Alive
Cookie: B=fn50ehd2612o2&b=3&s=rp; YMBM=d=&v=1;
```

Application: mail/yahoo/login
Fingerprint: mail/yahoo/login mail/arabic mail/yahoo/ymbm

TOP SECRET//COMINT//REL TO USA, FVEY

Sometimes the GENESIS programming language, which largely relies on Boolean logic, regular expressions and a set of simple functions, isn't powerful enough to do the complex pattern-matching required to detect certain types of traffic. In these cases, as one slide puts it, "Power users can drop in to C++ to express themselves." AppIDs or fingerprints that are written in C++ are called microplugins.

Here's an example of a microplugin fingerprint for "botnet/conficker_p2p_udp_data," which is tricky botnet traffic that can't be identified without complicated logic. A botnet is a collection of hacked computers, sometimes millions of them, that are controlled from a single point.

One document from 2009 describes in detail four generations of appIDs and fingerprints, which begin with only the ability to scan intercepted traffic for keywords, and end with the ability to write complex microplugins that can be deployed to field sites around the world in hours.

If XKEYSCORE development has continued at a similar pace over the last six years, it's likely considerably more powerful today.

—

Illustration for The Intercept by Blue Delli quanti

Documents published with this article:

- [Advanced HTTP Activity Analysis](#)
- [Analyzing Mobile Cellular DNI in XKS](#)
- [ASFD Readme](#)
- [CADENCE Readme](#)
- [Category Throttling](#)
- [CNE Analysis in XKS](#)
- [Comms Readme](#)
- [DEEPDIVE Readme](#)
- [DNI101](#)
- [Email Address vs User Activity](#)
- [Free File Uploaders](#)
- [Finding and Querying Document Metadata](#)
- [Full Log vs HTTP](#)
- [Guide to Using Contexts in XKS Fingerprints](#)
- [HTTP Activity in XKS](#)
- [HTTP Activity vs User Activity](#)

- [Intro to Context Sensitive Scanning With XKS Fingerprints](#)
- [Intro to XKS AppIDs and Fingerprints](#)
- [OSINT Fusion Project](#)
- [Phone Number Extractor](#)
- [RWC Updater Readme](#)
- [Selection Forwarding Readme](#)
- [Stats Config Readme](#)
- [Tracking Targets on Online Social Networks](#)
- [TRAFFICTHIEF Readme](#)
- [Unofficial XKS User Guide](#)
- [User Agents](#)
- [Using XKS to Enable TAO](#)
- [UTT Config Readme](#)

DONATE →



- [VOIP Readme](#)
- [Web Forum Exploitation Using XKS](#)
- [Writing XKS Fingerprints](#)
- [XKS Application IDs](#)
- [XKS Application IDs Brief](#)
- [XKS as a SIGDEV Tool](#)
- [XKS, Cipher Detection, and You!](#)
- [XKS for Counter CNE](#)
- [XKS Intro](#)
- [XKS Logos Embedded in Docs](#)
- [XKS Search Forms](#)
- [XKS System Administration](#)

- [XKS Targets Visiting Specific Websites](#)
- [XKS Tech Extractor 2009](#)
- [XKS Tech Extractor 2010](#)
- [XKS Workflows 2009](#)
- [XKS Workflows 2011](#)
- [UN Secretary General XKS](#)

WAIT! BEFORE YOU GO on about your day, ask yourself: How likely is it that the story you just read would have been produced by a different news outlet if The Intercept hadn't done it?

Consider what the world of media would look like without The Intercept. Who would hold party elites accountable to the values they proclaim to have? How many covert wars, miscarriages of justice, and dystopian technologies would remain hidden if our reporters weren't on the beat?

The kind of reporting we do is essential to democracy, but it is not easy, cheap, or profitable. The Intercept is an independent nonprofit news outlet. We don't have ads, so we depend on our members to help us hold the powerful to account. Joining is simple and doesn't need to cost a lot: You can become a sustaining member for as little as \$3 or \$5 a month. That's all it takes to support the journalism you rely on.

We're independent of corporate interests. Will you help us?

\$5	\$8	\$10	\$15
ONE-TIME	MONTHLY		
Become a Member →			